# Evolutionary Design of Electronic Devices and Circuits[±]

**Adrian Stoica, Gerhard Klimeck, Carlos Salazar-Lazaro, Didier Keymeulen[1], and Anil Thakoor**
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
adrian.stoica@jpl.nasa.gov

**Abstract- This paper addresses the use of evolutionary algorithms in the design of electronic devices and circuits. In particular, the paper introduces the idea of evolutionary design of nanodevices, and illustrates it with the design of a resonant tunneling diode. A second experiment, this time using CMOS microdevices, illustrates the use of evolutionary algorithms for circuit design. The experiments were facilitated by an Evolutionary Design Tool developed around a parallel implementation of genetic algorithms (using PGAPack), and device/circuit simulators (NEMO and SPICE). It is speculated that in the future devices and circuits may be simultaneously co-designed.**

## 1 Introduction

### 1.1 Design of new devices and adaptive circuits

The NASA goal to reduce payload in future space missions by avoiding on-off and artisanal task while increasing mission capability demands miniaturization. The miniaturization is obtained by applying nano-technologies which may result in devices such as application-specific integrated microinstruments and nano-satellites (satellites weighing only a few kilograms). While silicon micro device technology dominates the commercial microprocessor and memory market, the semiconductor heterostructure nano-devices allows to build miniature 2-D and 3-D sensors, actuators and intelligent microinstruments applied for example for light detection, light emission, and high-speed data transmission.

The design of nanoelectronics still faces major challenges, both at device and circuit level. Although the production of these heterostructure nano-devices is enabled by the advancement of material growth techniques, the dimension reduction change the importance of some effects with respect to ordinary-sized components and some characteristics of the materials used introduce undesired effects must be compensated by new designs. The full experimental exploration of this design space is unfeasible and a reliable design tool is needed. Here is where a device simulator tool plays a paramount role. Even more, such a tool can be combined with an automated search tool, such as an evolutionary design tool, to assist in design; this largely enpowers the human designer. Beyond devices, circuits based on nanodevices have a great potential, especially in the area of fast switching logic; such designs are still a largely untouched area.

Not only nanoelectronics, but microelectronics can also largely benefit from automation of design. As it is often said, analog design is more an art than a science, and the number of experts in the field is really smaller than the need, especially in the context of RF portable devices and mixed-signal circuits in systems-on-a-chip solutions.

This paper presents a general tool based on evolutionary algorithms which facilitates evolutionary design of electronics, and illustrates the evolutionary design of a nanodevice and of a microelectronic circuit.

### 1.2 Evolutionary algorithms in electronic design

The idea behind evolutionary design of electronic devices and circuits is to employ a search/optimization algorithm that operates in the space of all possible devices/circuits and determines a solution that meets the required specifications [1], [2], [3]. Most experiments focused on evolutionary design of circuits and used either Genetic Algorithms (GA) or Genetic Programming (GP). Evolutionary design of nano-devices follows the same pattern as the evolutionary design of circuits, and a same evolutionary design tool may be used. This section illustrate the concept using the evolutionary design of circuits.

The evolutionary/genetic search is tightly coupled with a coded representation for the circuits. Each circuit gets associated a "genetic code" or *chromosome;* the simplest representation of a chromosome is a binary string, a succession of 0s and 1s that encode a circuit. Synthesis is the search in the chromosome space for the solution corresponding to a circuit with a desired functional response. The genetic search follows a "generate and test" strategy: a population of candidate solutions is maintained

[±] In *Proceedings of the 1999 Congress on Evolutionary Computation*, July 6th-9th, 1999, Washington DC. IEEE Press.

[1] Also member of the Electrotechnical Laboratory, Tsukuba, Japan.

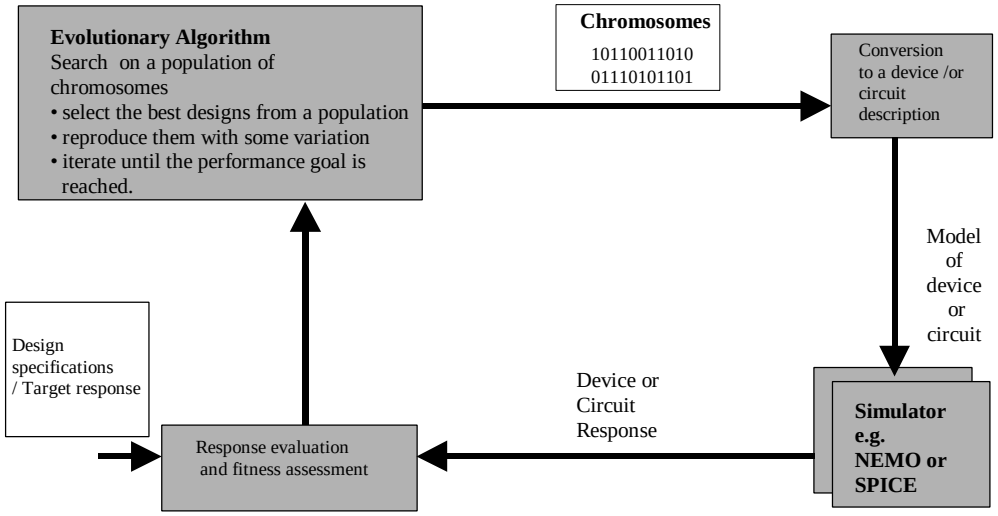



**Figure 1:** Evolutionary design of electronic devices and circuits

at each time; the corresponding circuits are evaluated and the best candidates are selected and reproduced in a subsequent generation, until a performance goal is reached. The main steps of evolutionary design are illustrated in Figure 1. First, a population of chromosomes is randomly generated. The chromosomes are converted into device models parameters (for device design) or circuit models (for design of circuits). Circuit responses are compared against specifications of a target response, and individuals are ranked based on how close they come to satisfying it. Preparing for a new iteration loop, a new population of individuals is generated from the pool of best individuals in the previous generation, some individual being taken as they were and some being modified by genetic operators, such as chromosome crossover and mutation. The process is repeated for many generations, and results in increasingly better individuals. The process is usually stopped after a number of generations, or when the closeness to the target response has reached a sufficient

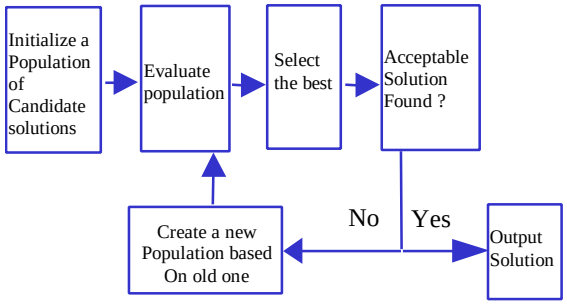



**Figure 2**: Sketch of a simple GA

degree. One or several solutions may be found among the individuals of the last generation.

## 2 An Evolutionary Design Tool

A variety of Evolutionary Algorithms have been used successfully for evolution of circuits. A GA was chosen here because (a) previous work has demonstrated its efficiency in evolutionary circuit synthesis, (b) the mechanism is simple to understand and implement, (c) public domain software exists and saves development time, and (d) the focus was on the reconfigurable hardware (in the case of circuits) and not on the reconfiguration mechanism. It is likely that more intelligence can be inserted into the search mechanism. A simple block diagram of operations taking place in the GA is illustrated in Figure 2.

An evolutionary design tool was built to facilitate experiments in simulated evolution. The tool can be used for synthesis and optimization of new devices, circuits, or architectures for reconfigurable hardware. These operations get performed before the mission and before any hardware gets fabricated. The tool can also be used in hardware-software co-design before the mission. In its current implementation the tool uses the public domain Parallel Genetic Algorithm package PGAPack and two simulators, the Nanoelectronic Modeling Tool (NEMO) and SPICE. An interface code links the GA with the simulator where potential designs are evaluated, while a graphical user interface facilitates the formulation of requirements and visualization of results. Each generation the GA produces a new population of binary chromosomes, which get converted into structural parameters that enter device models, or voltages in Spice netlists that describe candidate circuit designs. The devices specified by the parametric models are simulated by NEMO. The circuits expressed by netlists are simulated by a public domain version of SPICE

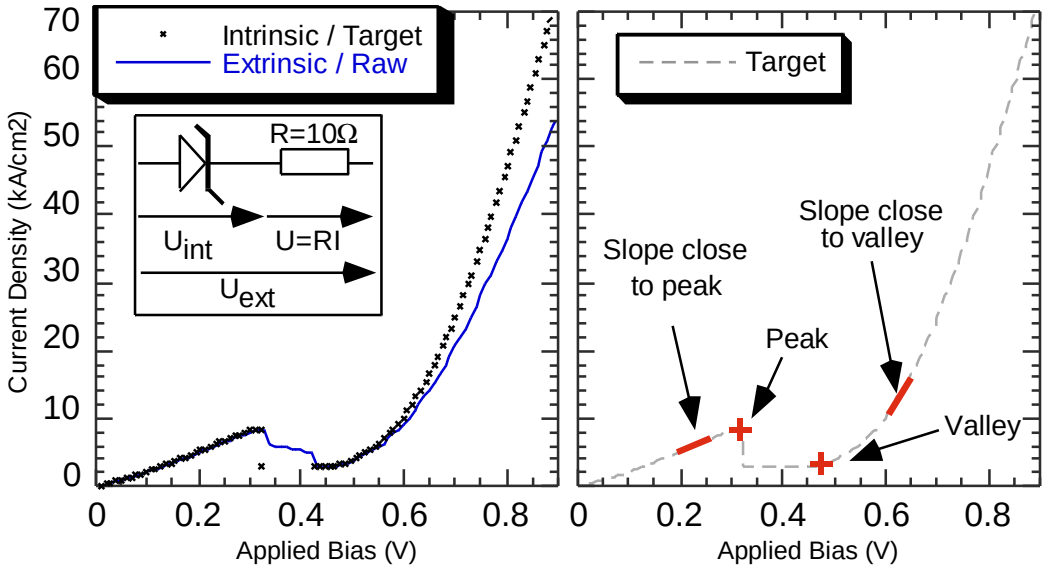



**Figure 4**: Generation of the target I- characteristic of a typical resonance tunneling diode. (a) The extrinsically measured I-V (solid line) includes a series resistance and oscillations in the negative differential conductance region (0.32V-0.43V). The series resistance can be estimated from a series of devices with different cross sections. The intrinsic I-V is the target for the optimization (crosses). (b) Features that enter into the evaluation of the fitness of simulated data. Of particular interest are the peak and valley voltage and current and the slopes close to the peak and the valley.

3F5 as the circuit simulator. The tool was used on a 256-processor machine to simulate evolution.

## 3 Evolution of nanoelectronic devices and circuits

The term nanotechnology is broadly deferred to the synthesis and the integration of materials and process devices at the level of molecule. One of the key point of the nanotechnology is to exploit the material variations on an atomic scale which enable the quantum mechanical functionality of devices such as resonant tunneling diodes (RTDs), quantum well infrared photodetectors (QWIPs), quantum well lasers, and heterostructure field effect transistors (HFETs). The design and optimization of such

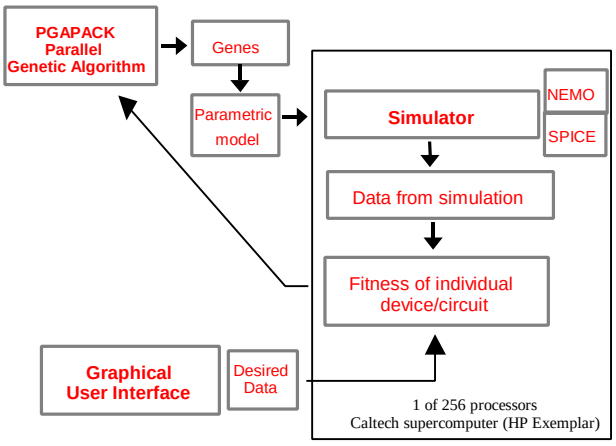



**Figure 3:** An Evolutionary Design Tool

heterostructure devices requires a detailed understanding of quantum mechanical electron transport. NEMO is a general-purpose quantum device design and analysis tool that addresses this problem. NEMO was developed as a general-purpose quantum mechanics-based 1-D device design and analysis tool from 1993-97. The tool is available to US researchers by request on the NEMO web site [4]. NEMO is based on the non-equilibrium Green function approach, which allows a fundamentally sound inclusion of the required physics: bandstructure, scattering, and charge self-consistency. The theoretical approach is documented in references [5, 6] while some of the major simulation results are documented in references [7-9].

Heterostructure device designs involve the choice of material compositions, layer thicknesses, and doping profiles. Material parameters such as band offsets, effective masses, dielectric constants etc. influence the device simulation results in addition to the structural design parameters. The full exploration of the design space using purely experimental techniques is unfeasible due to time and financial constraints. For example, it takes a well-equipped research laboratory approximately five working days [10] for the growth, processing and testing of a particular resonant tunneling diode design. NEMO can provide quantitative [7-9] current voltage characteristics (I-V's) within minutes to hours[2] of CPU time for a single set

[2] The actual CPU time needed for a single I-V simulation depends strongly on the choice of material systems, bandstructure models, temperature scattering models, and

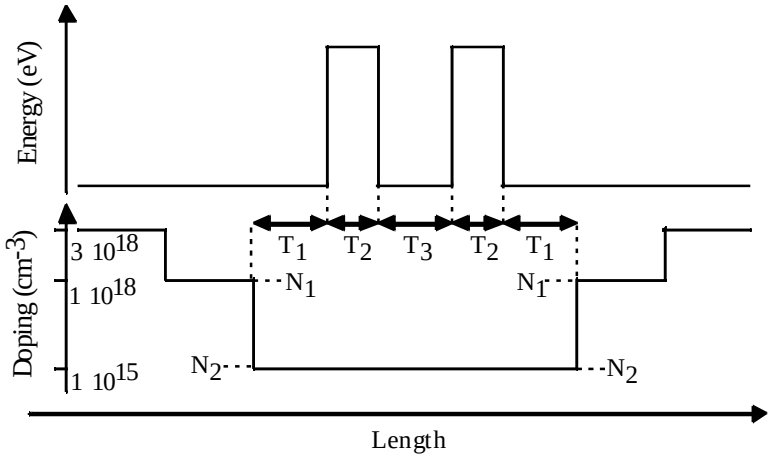



**Figure 5.** Conduction band edge and doping profile of a typical resonant tunneling diode. The central device region is typically undoped. The low doped spacer thickness, the barrier thicknesses and the well thickness are labeled $T_1$, $T_2$, and $T_3$, respectively. The low spacer doping and the central device doping are labeled $N_1$ and $N_2$, respectively. These five parameters are varied.

of device and material parameters. With this quantitative simulation capability the design parameter space can be explored expediently once an automated system for the design parameter variation is implemented. The architecture lends itself to the optimization of any parameters that enter a NEMO simulation. To evaluate how good a particular parameter set is, a fitness function must be developed.

### 3.1 Simulation Target and Fitness Function

In this work the RTD is used as a vehicle to study the effects of structural and doping variations on the electron transport. Current voltage characteristics of two devices that are part of a well-behaved test matrix of experimental data published in reference [8] are used as a design target. The raw I-V data (see the example in Figure 4) contains a contact series resistance and oscillations in the negative differential resistance (NDR).

The fitness of the simulated data is measured against such target I-V. There are four particular features that are explicitly evaluated for each simulated I-V: peak and valley current and voltage, and the slope close to the peak and the valley (see Figure 4b). Differences between the target and the simulation in these four features and the absolute and relative error for all simulated data points enter into the fitness function with a weighted average. The fitness values between 0 and 1. The electron transport simulations are based on a single band model, which incorporates effects of non-parabolic bands in the longitudinal and transverse directions relative to electron transport [6].

bias points. The individual I-V characteristics presented here take about 30 minutes to compute on a single 200MHz R10000 CPU of an SGI Origin.

### 3.2 Set-up of Numerical Experiment

In the numerical experiment described in Figure 5, five parameters (2 doping concentrations, $N_1$, $N_2$, and 3 thicknesses, $T_1$, $T_2$, $T_3$) are varied within the genetic algorithm in order to achieve the best fit to an experimental I-V curve. The simulation is started from a random population of 200 parameter sets. The doping population is logarithmically distributed around the nominal values by factors of 10 ($N_1$? [$1x10^{17}$,$1x10^{19}$], $N_2$? [$1x10^{14}$,$1x10^{16}$]). The layer thickness population is uniformly distributed around the nominal value by 10 monolayers ($T_1$? [1,17] for device 1, $T_1$? [10,30] for device 2, $T_2$,$T_3$? [6,26]). In each generation 63 of the worst genes[3] are dropped out of the population and new genes are generated [11] from the rest by mutation and crossover.

### 3.3 Simulation Results

Two I-V's from slightly different structures serve as a target of the genetic algorithm optimization. Both structures were specified to the grower to have 16 monolayers (ml) of barriers ($T_2$) and well ($T_3$), no intentional doping in the central device ($N_2$=$1x10^{15}$ $cm^{-3}$), $N_1$=$1x10^{18}$ $cm^{-3}$ doping in the low doping spacers, and $3x10^{18}$ $cm^{-3}$ in the high doping contacts (see Figure 5). The nominally only difference in the two devices is in the no-doping spacer length $T_1$ of 7 vs. 20 ml. The simulation is started from the random populations as described in the previous section. The genetic algorithm converges for both I-V's to the nominal

[3] LAPACK is implemented with MPI where N-1 of N processors are slaves to one master processor. The master takes care of the collection of data from the slaves. In a cluster of 64 CPU's we therefore renew only 63 genes in every generation.

structure values, well within the experimental uncertainty as shown in Figure 6. Again it is found that the well widths must be increased in the simulation by a few monolayers versus the nominal values to achieve the best agreement with experimental data [8]. Different relative weights will result in different "optimal" structures.

### 3.4 Discussion

This work is the first step to integrate NEMO within a high performance parallel computational environment. A

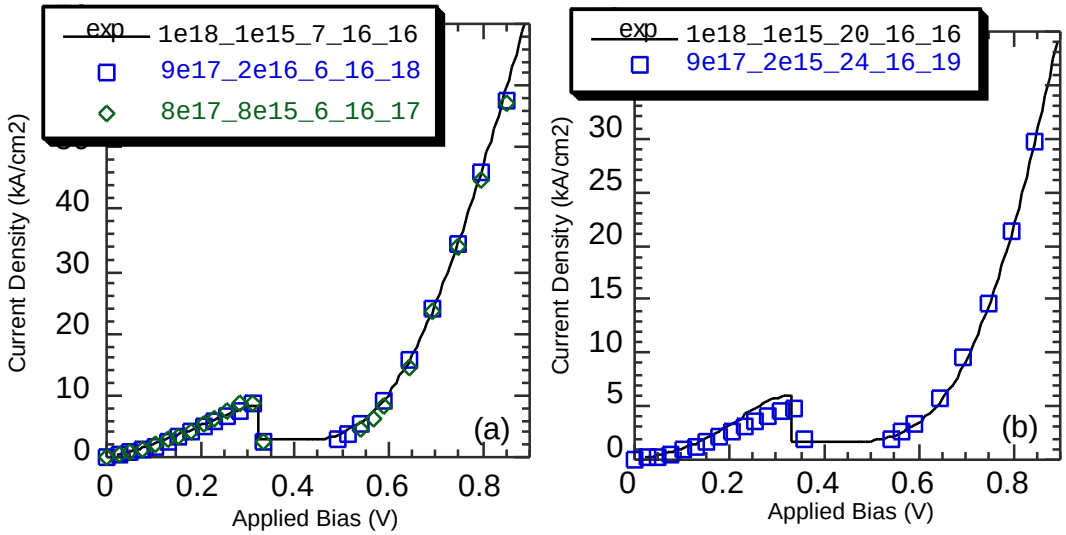



**Figure 6**: Current voltage characteristics of two different InGaAs/InAlAs resonant tunneling diodes. The nominal structures have barrier ($T_2$) and well ($T_3$) thicknesses of 16 monolayers (ml), and doping a doping profile of $10^{18}$ $cm^{-3}$ ($N_1$) and $10^{15}$ $cm^{-3}$ ($N_2$). The devices (a) and (b) differ nominally in their no-doping spacer thicknesses ($T_1$) of 7 and 20 ml, respectively. The solid lines show experimental data published in reference [5], where the noise in the valley current region was eliminated. The curves are labeled by the 5 parameters N1_N2_T1_T2_T3.

desired curve can now be entered as the target of the simulation and the genetic algorithm is expected to obtain the optimal parameter set. Future work will utilize this method to analyze the vast material and structure parameter space. It is planned to evaluate other optimization techniques such as simulated annealing and directive approaches as well. These optimization techniques will be made available within a graphical user interface which enables the selection of parameters to be varied, the setting of parameter ranges and the setting of optimization parameters, such as population sizes, and mutation and crossover rules.

## 4 Evolution of microelectronics

This section introduces evolution of CMOS circuits based on Programmable Transistor Arrays, describing a design for hardware reconfigurable at transistor level. The PTA allows synthesis of analog, digital and mixed-signal circuits, being a more suitable platform for synthesis of analog circuitry than existing FPGAs or FPAAs, extending the work on evolving simulated circuits to evolving analog circuits directly on the chip.

### 4.1 Programmable Transistor Array

The idea of PTA as a platform for evolutionary microelectronics was introduced in [18], and expanded in [19]. The proposed PTA is an array of transistors interconnected by programmable switches. The status of the switches (On or Off) determines a circuit topology and consequently a specific response. Thus, the topology can be considered as a function of switch states, and can be

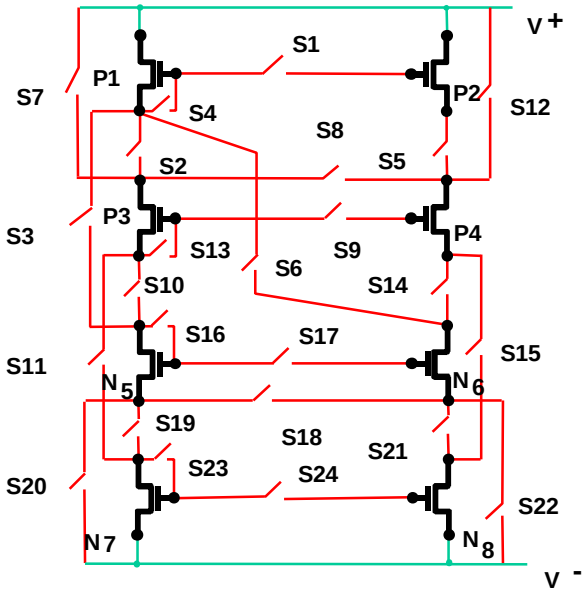



**Figure 7**: Module of the Programmable Transistor Array

represented by a binary sequence, such as "1011…" , where by convention one can assign 1 to a switch turned On and 0 to a switch turned Off. The PTA is a modular architecture, in which modules can be cascaded to determine a more complicated circuit topology. Figure 7 illustrates an example of a PTA module consisting of 8 transistors and 24 programmable switches. In this example the transistors P1-P4 are PMOS and N5-N8 are NMOS, and the switch based-connections are in sufficient number to allow a majority of meaningful topologies for the given transistors arrangement, and yet less than the total number of possible connections.

Programming the switches On and Off determines a circuit for which the effects of non-zero, finite impedance of the switches can be neglected in the first approximation. An example of a circuit drawn with this simplification is given in Figure 8.

The left drawing illustrates the ideal circuit, the right drawing shows with dotted lines the finite resistance of open switches. A power supply, input signals and a biasing current source have been added.

In this implementation four layers of transistors (two PMOS and two NMOS) were chosen, but this can be increased, for example to 6 or 8. On the "horizontal" direction the PTA architecture allows implementing bigger circuits by cascading PTA modules. A simple expansion

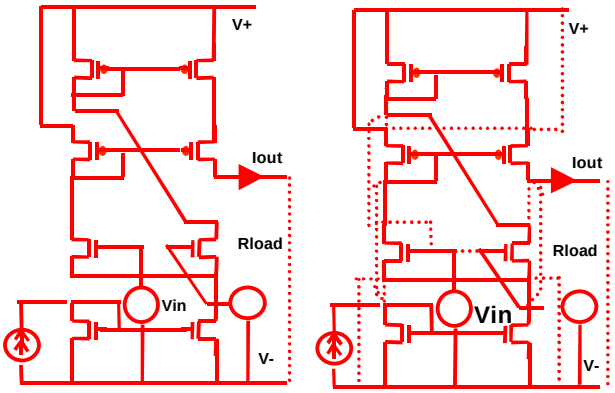



**Figure 8:** Schematic of a simple circuit implemented on the PTA module (with finite resistance of Off switches as dotted lines on the right figure)

would be by connecting two adjacent modules with a set of programmable connections. One such expansion with 24 connections between two modules (and thus a total of 72 programmable elements) was simulated. More details on the hardware aspects are discussed in [20].

### 4.2 Evolution of a Gaussian Circuit

The evolutionary synthesis approach illustrated in Figure 1 was applied to the model of PTA illustrated in Figure 7. Evolution of a computational circuit was chosen to illustrate the approach. The goal of evolution was to synthesize a circuit which exhibits a Gaussian I-V characteristic. In a previous experiment [21] the circuit topology was fixed and the search search/optimization addressed transistor parameters (channel length and width); such evolution proved quite simple. The search for a topology turned out to be a much harder problem and several architectures were unsuccessfully attempted before the PTA was conceived. In the PTA case, the transistor parameters were kept fixed and the search was performed for the 24 binary parameters characterizing switches status. The specification of the fitness function played an important role in accelerating evolution. A simple Euclidian distance was found much less efficient than a fitness function based on a weighted combination of parameters that specify distances to some control points such as illustrated in Figure 9. The evolution was simulated on a Caltech supercomputer (HP-Exemplar), using the Evolutionary Design Tool. Successful evolution was demonstrated on multiple runs with populations between 50 and 512, evolving for 50 or 100 generations. The execution time depends on the above variables and on the number of processors used (usually 64 out of the 256 available), averaging around 20 minutes (the same evolutions took about 2 days on a SUN SPARC 10). In some runs the solution circuit shown in Figure 8 (human designed) was rediscovered by evolution.

Other solutions found include the circuits illustrated in Figure 11, which produce the first two responses in Figure 10; some other responses from the same generation are illustrated in Figure 10 for comparison.

### 4.3 Discussion

It is interesting to analyze in more detail the unusual solutions found by evolution. Circuits like those illustrated in Figure 11 resulted from evolutionary similar (under certain test conditions) to that of the circuit shown in Figure 8.Thicker dotted lines show connections that existed in the circuit in Figure 5, but are missing in the circuits in

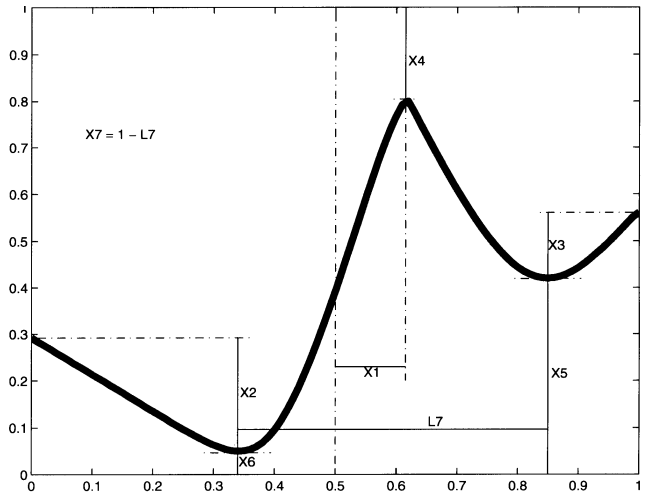
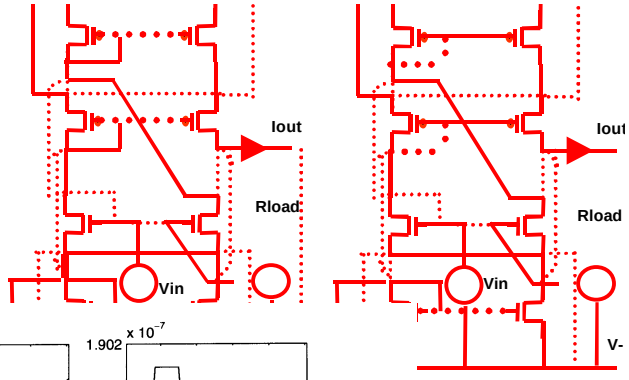
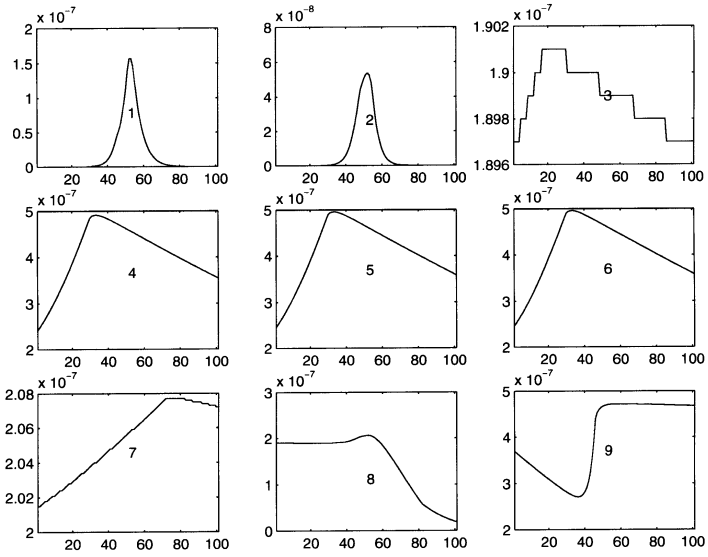



**Figure 9:** Parameters used for the specification of the fitness function. Fitness =f(x1 ... x7)

Figure 11. As it is easy to observe these circuits are outside normal design practices, e.g., the transistors P2, P4 and N8 on the left circuit in Figure 11 and P2 have floating gates. The reality is that the switches have a big, but finite,




ion; their design is tices.

**Figure 10**: Best circuit responses in a simulated evolution

resistance in the Off state (~MOhms or GOhms) and a non-zero resistance/impedance in the On state (~ tens of Ohms). *One observation from here is that while the effects of non-perfect switches may be negligible in a first approximation for many digital circuits, such effects may fundamentally affect analog programmable circuits.*

## 5 Toward device/circuit co-design?

Current electronic design methodology is based on using components with pre-specified, well understood characteristics and combining them in arrangements that give the ensemble a desired/target functionality. Circuit solutions have been over the years collected in design handbook and the designer is many times in the position of calculating component values, running analysis and comparing results, etc. If one would put now designers to design with circuits with weird devices, say with 4 peaks of the I-V characteristic they will be quite puzzled. But in fact it may well be that the most efficient implementation in silicon of a circuit would be by using 2 devices of 4-peak nature. Unlike human designers, evolution is not puzzled and can perfectly well design with wired devices. In fact the device characteristic and the circuit topology may be simultaneously determined for optimal silicon implementation and system performance. Thus, it is possible that the future tools for automated design will perform simultaneous device design and circuit design.

## 6 Conclusion

An Evolutionary Design Tool was developed around PGAPack, with NEMO and SPICE as simulators, to facilitate evolutionary design of devices and circuits. A nanoelectronic device (resonant tunneling diode) with a desired current-voltage characteristic was obtained by evolutionary design, following an optimization of five structural parameters such as layer thickness and doping profiles. The convergence of the initially random population of devices to experimental specified device parameters is demonstrated for two different devices. A circuit implemented on a Programmable Transistor Array topology was also synthesized to provide a target response. The effect of non-ideal switches used in evolvable/reconfigurable analog circuits appears to play an important role in the final circuit design.

## Acknowledgements

The research described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics and Space Administration.

## Bibliography


[1] Ning, Z-Q, Mouthaan, T. and Wallinga, H. SEAS: A Simulated Evolution Approach for Analog Circuit Synthesis. In *Proceedings of the IEEE 1991 Custom Integrated Circuits Conference*, pp 5.2.1-5.2.4. May 12-15, 1991, San Diego, USA. IEEE Press: Piscataway, NJ.

[2] De Garis, H. Evolvable Hardware: Genetic Programming of a Darwin Machine. In *Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms*, 1993, Innsbruck, Austria. Springer Verlag: Berlin.

[3] Higuchi T. et al. Evolvable Hardware with Genetic Learning: A first step towards building a Darwin machine. In J-A. Meyer, H.L. Roitblat and S.W. Wilson (eds.), *Proceedings of the 2nd International Conference on the Simulation of Adaptive Behavior*, pp 417-424, 1992. MIT Press: Cambridge.

[4] NEMO, Nanoelectronic Modeling, in http://www.raytheon.com/rtis/nemo/.

[5] Lake R. et al.. In *Journal of Applied Physics*, **81**(12), pp. 7845, 1997.

[6] R. Lake et al.. In *Journal of Physical Static Solid*, (b), **204**, pp. 354, 1997.

[7] G. Klimeck et al.. In *Applied Physic Letter*, **67**(17), pp. 2539, 1995.

[8] G. Klimeck et al..In *IEEE DRC*, p. 92, 1997.

[9] R. C. Bowen et al.. In *Jorunal of Applied Phys*ics, **81**, pp. 3207, 1997.

[10] A. C. Seabaugh. Texas Instruments, private communication, 1997.

[11] D. Levine, http://www-unix.mcs.anl.gov/~levine/PGAPACK/index.html, Parallel Genetic Algorithm Library.

[12] J. N. Schulman. *Second Workshop on Characterization, Future Opportunities and Applications of 6.1Å III-V Semiconductors*, Aug. 24-26, 1998, Naval Research Laboratory, Washington, DC, http://estd-www.nrl.navy.mil/code6870/code6870.html.

[13] H. C. Liu. In *Journal of Applied Phys*ics. **64**, pp. 4792, 1988.

[14] H. C. Liu. In *Journal of Applied Physics.* **53**, pp. 485, 1988.

[15] J. Koza, F.H. Bennett, D. Andre, and M.A Keane. Automated WYWIWYG design of both the topology and component values of analog electrical circuits using genetic programming. In *Proceedings of Genetic Programming Conference*, pp. 28-31, 1996. AAAI Press.

[16] J. Lohn, J. and S. Colombano. Automated Analog Circuit Synthesis using a linear representation. In M. Sipper, D. Mange and A. Perez-Uribe (Eds*) Evolvable Systems: From Biology to Hardware*, pp. 125-133. ICES'98, Lausanne, Switzerland. Springer-Verlag, Lecture Notes in Computer Science 1478, Berlin.

[17] A. Thompson. An evolved circuit, intrinsic in silicon, entwined in physics. In T. Higuchi, M. Iwata and W. Liu (eds.), *First International Conference on Evolvable Systems*, pp.390-405. Springer-Verlag, Lecture Notes in Computer Science 1259, 1996.

[18] Stoica, A.. *Reconfigurable Transistor Array for Evolvable Hardware*, July 26, 1996. Caltech/JPL Novel Technology Report.

[19] Stoica, A. and Salazar-Lazaro, C. *Evolutionary technique for automated synthesis of electronic circuits*, September 4, 1998. Caltech/JPL Novel Technology Report.

[20] Stoica, A. Toward Evolvable Hardware Chips: Experiments with a Programmable Transistor Array. In *Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-inspired Systems*, Microneuro'99, Granada, Spain, April 7-9, 1999.

[21] Stoica, A. On hardware evolvability and levels of granularity *International Conference on Intelligent Systems and Semiotics*, NIST Gaithersburg VA, September 1997.